Source Discovery and Content Matching in Heterogeneous Information Environments

A Dissertation Proposal Presented to the

Department of Computer Science

Brigham Young University

In Partial Fulfillment of the Requirements

for the Degree of Doctor of Philosophy

Li Xu

July, 2001

1 Introduction

In open and evolving environments such as the World Wide Web, the amount of information available is proliferating at a tremendous rate. Data on the Web comes in various forms. At one extreme, online traditional database systems manage large amounts of highly structured data, much of it behind form interfaces. At the other extreme, many Web pages are totally unstructured (containing raw text written in natural language), or at least only informally semistructured (containing some elements of typed or structured data presented in an unstructured or semistructured way). Between these two extremes, the most important of the various forms is XML, which provides a way to formally model semistructured data.

With all of this data, one of the huge challenges we face is the integration of information that is extracted from structured, semistructured, and unstructured sources pertaining to a domain of interest. Our approach to this problem begins with a domain-of-interest specification, which we call a *target view.* We define a target view as a conceptual schema like the ontology in [ECJL+99]. Such an ontology consists of a description of object sets and relationship sets among the object sets that are of interest to the end user (the creator of the ontology). An ontology may also include constraints and reasoning or inference rules that further describe the domain of interest. Given a particular target view, we can integrate multiple information sources by (1) discovering applicable information sources; (2) individually matching and reconciling the concepts in applicable information sources with corresponding concepts in the target view; and (3) merging the related data. The target view is a central, organizing concept for our approach.

The objective of this dissertation is to resolve the problems of discovering applicable information sources, and then matching and reconciling information sources with a target view[1]. Significantly, a target view is independent of any particular information source.

Given a target view, we address the following two sub-problems:

(1) Source discovery — we first determine whether an information source or any portion of the source is applicable to the target view.
(2) Source matching and reconciliation — given a set of applicable sources, it is likely that individual sources will use different object sets, different terminology for the same object sets, different relationship sets, or different values with the same object sets. Therefore we must match elements of the various applicable sources with the target view, and reconcile these differences.

Solving these two problems will allow us to identify data of interest and bring it into a target view, a common formalism, so that the data can be fully integrated (assuming that the downstream problem of merging related data is resolved). Before giving the details

[1] These problems are difficult enough that the downstream problem of merging related data is deemed to be beyond the scope of this dissertation.

about how we intend to solve those two problems, we first give an overview of related work in source discovery and information integration.

1.1 Source Discovery Overview

Discovering desired information is related to information retrieval (IR) [FB92, SM83]. IR is concerned with indexing a collection of documents based on document topics, and answering queries by returning a ranked list of relevant documents [BR99]. In particular, the focus is on measuring the similarity of two documents or the relevance of documents to a given request. In contrast, the central focus for discovering desired information in our approach is to identify documents that contain objects and relationships compatible with those specified by a desired target view. Thus, the discovered information not only relates to the topic of a particular application, which is similar to the goal of the IR problem, but also relates to the structure of the target view. Moreover, the object sets and relationship sets comprise a finer level of information than the application topics used in the IR approach. Given this finer level of information structure, the data contained in an information source can be extracted according to the specification in a target view.

The classical approach of IR is to rely on keyword matching. That is, keywords are extracted from each document either manually or automatically, and documents are compared based on the extracted keywords. Interrelationships among keywords are usually ignored. For example, while a classical IR approach might classify one document containing a sentence like "Wilhemine Friderike Carolina ERDMAN, born on 30 OCT 1864, Buffalo, Erie, NY…", it will not consider the relationship between "Wilhemine Friderike Carolina ERDMAN" and "Buffalo, Erie, NY". Thus, classical IR systems cannot precisely find applicable information sources that match a target view containing relationships like "person was born at place". This is because keywords do not indicate which object set patterns and relationship set patterns are compatible with the specifications declared in a target view. Classical IR systems are good for preliminary exploration, but they retrieve too many irrelevant information sources for our purposes.

We can provide a better source discovery solution by using all the information in the target view specification. Given the values recognized by the target view specification and metadata extracted from an information source, we can construct a set of applicability heuristics. Then, based on machine-learned rules over these heuristic measurements, we determine whether an information source is applicable to a given target view. Our approach can markedly improve precision over the keyword-based IR approaches.

Our approach for source discovery largely utilizes machine-learning techniques to exploit a broad set of heuristics to produce rules to detect applicable information sources. Modern IR systems also include learning-based text classification [BB63, Hoyle73, Maron61] as an IR task. Text classification often depends on keyword-based techniques and statistical learning methods to classify document topics. While applying text classification methods in IR systems, one or more category labels are assigned to a document. This method presumes a predefined, static set of user interests (categories) but suffers because of a lack of semantic analysis. Compared with our approach, text classification methodology cannot process complex information requests specified as target views.

Information extraction techniques based on natural language processing (NLP) have been shown to be effective for high precision text classification [RL94], which is able to deal with more complex information requests than traditional text classification. However, NLP techniques are computationally expensive, especially for user requests as complex as the specifications declared in a target view. Thus, it is difficult for NLP approaches to scale up to the large quantity of documents in the context of our source discovery and information integration problem. In contrast, our approach depends on ontology-based data extraction techniques [ECJL+99] to recognize values from an information source, so our data-extraction technique is not as expensive as NLP alternatives.

Ontology-based data extraction techniques maintain the knowledge for an application domain in the specification of a desired target view. Similarly, within the IR community, knowledge-based IR systems [Goodman91, HW91, RJ91] have been developed that address some problems raised in keyword-based IR systems using rule bases or other knowledge resources. Knowledge-based IR systems have attempted to capture searchers' and information specialists' domain knowledge and classification scheme knowledge. However, they do not usually have learning ability and only perform what they are programmed to do. Moreover, most knowledge-based IR systems require an extensive manual knowledge-engineering effort that takes significant time and human resources to acquire knowledge from domain experts. In contrast, our approach has the ability to learn the knowledge needed to do the classification and thus is much easier to port to new application domains**.**

1.2 Information Integration Overview

In our approach, we attack the problem of source matching and reconciliation in the context of information integration systems. In the following, we give an overview of information integration systems as well as approaches to the problem of source matching and reconciliation.

From a practical view of point, an important distinction in building an information integration system is whether to take a warehousing or a virtual approach [HZ96, Hul97]. In the warehousing approach [ZGHW95, LMSS95], data from multiple information sources is loaded into a warehouse, and user queries are applied to the data warehouse. The warehouse approach requires updates to the warehouse when the source data changes and the updates are typically done in batches, not on demand. Such an approach guarantees adequate query performance because queries are read only and operate on a single repository. In the virtual approach, the data remains in the information source, and user queries are decomposed at run time into queries on the information sources. The virtual approach is appropriate when the number of information sources is large and individual information sources change frequently, but it requires more sophisticated query optimization and execution methods to guarantee adequate performance.

From a theoretical point of view, the traditional approach to information integration is to create a mediated schema, which is a set of object sets and relationship sets that describe individual entities and relationships between the entities in the domain of expected user queries. To evaluate queries, the information integration system translates queries on the mediated schema into queries on the underlying information sources. There are two sub-approaches [Ullman97]: global-as-view (GaV) [BGLM+99,

CHSA+95, GPQR+97, Hammer99, IFF99, LAW, Singh98, TRV96,] and local-as-view (LaV) [ACHK93, FPNB99, GBMS99, GKD97, LRO96, MKSI96, PSBG+99, SSR94].

In the GaV approach, global concepts, which consist of object sets and relationship sets found in the sources, are synthesized in the mediated schema. In the LaV approach, global concepts are declared independent of information sources. The main advantage of the GaV approach is that query reformulation is simple, because it reduces to view unfolding, which means that queries to the sources are immediately available based on the view specification. However, whenever a new information source is added or a current underlying information source is changed, the change needs to be propagated to the mediated schema. In contrast, in the LaV approach, it is simple to add or delete information sources since the source descriptions do not need to take into account the interactions with other source descriptions. But it is difficult for a LaV information integration system to process complicated queries because the LaV approach has to apply query rewriting algorithms to decompose a user query into subqueries in terms of the information sources. The query rewriting algorithms generate the rewriting in time that is exponential in the size of the query.

Our target-view approach complements both the GaV and the LaV approaches. Like GaV, our target view provides a mediated schema in terms of global concepts and the interrelations between the concepts, but like LaV, the mediated schema is declared independent of the information sources. The independent target view limits the scope of the integration to a predefined set of concepts. This makes the integration work more manageable than the traditional GaV approach and keeps the scalability as in a LaV approach. Further, each information source has its own independent source view declaration, which is a populated conceptual schema. We map the target view to each source view in such a way that query formulation results in view unfolding. This allows the integration to have better query response time than the traditional LaV approach. The cost of our target-view approach is that we need manually construct a target view and produce a mapping between the target view and each of the information sources. Our experience in teaching others to construct target views suggests that a target view for an application such as automobile want-ads can be created in a few dozen person-hours.

Work on generating a mapping between a target view and an information source can be classified into rule- and learner-based matching approaches. Rule-based matching approaches usually utilize only schema information and normally only in a hard-coded way. Whenever the system changes to a new application domain, the hard-coded rules must be changed dependent on domain constraints. We use a machine-learning approach, which exploits a broad set of properties utilizing both schema and data-instance information. Based on the set of properties, we construct a set of similarity measurements between the object and relationship sets of the target view and the object and relationship sets of a source view. Then, based on machine-learned rules over these similarity measurements, we determine whether an object or relationship set (explicit or derived) in the source view can match an object or relationship set in the target view. Compared with rule-based matching [CA99, MBR01, MZ98, PSU98, e.g.], our approach is easier to port to a new application domain because the learning methods automatically generate the matching rules. Compared with other machine-learning approaches [CHR97, DDH01, LC00, MHH00, PE95], the most distinguished difference of our approach is that we construct a broad set of properties based on both schema and data-

instance information in a systematic way by utilizing the specifications of source views and target views, which are conceptual model instances modeled based on OSM [EKM92]. OSM is the formal foundation of our target-view ontologies.

2 Thesis Statement

We can resolve several problems with existing approaches to information discovery and integration by starting with a target view specified as an ontology that is conceptual and independent of any information source. We can develop semi-automatic procedures that use a given target view *T* to (1) identify whether an information source *S* applies to *T*, and (2) generate a mapping between *T* and *S*. Generated mappings allow data to be extracted from applicable sources so that queries processed against the target can be applied to a wide range of applicable sources. Our approach constitutes an improved solution because our method
(1) can process more complex user requests (specified in a single target view),
(2) can discover applicable information sources more precisely,
(3) is more scalable and portable than IR systems, and
(4) retains both the query performance of the GaV approach and the scalability of the LaV approach to unify a large number of heterogeneous information sources.

3 Research Description

This dissertation includes two components: one is to identify information sources applicable to a desired target view and the other is to generate target-to-source mappings between the concepts of the target view and the concepts of applicable source views. The issues associated with these two components are as follows: (1) applicability of information sources, (2) analysis of information sources, (3) direct semantic matches, (4) indirect semantic matches, (5) object-set similarity measurement, (6) relationship-set similarity measurement, and (7) implementation. The seven issues are addressed in the following discussion.

A target view is specified as an ontology (defined in [ECJL+99]), which consists of two components (1) an *object/relationship-model instance* that describes sets of objects, sets of relationships among objects, and constraint over object and relationship sets, and (2) for each object set, a *data frame* that defines the potential contents of the object set and may also include constraints and rules for reasoning and inferencing. A data frame for an object set defines the lexical appearance of constant objects for the object set and establishes appropriate keywords that are likely to appear with the object in an information source.

A source view is similar to a target view, and is an ontology constructed based on metadata extracted from an information source. However, in a source view, the data frames describe data values associated with the object sets in the underlying information source. We assume that we can construct the source views of the applicable information sources based on existing wrapper generation techniques (e.g. [Adelberg98, DDL00, ECJL+99, GLSR00, MMN99, SA99]).

The OSM conceptual model [EKW92] provides a uniform model for target views and source views. There is a graphical version and a textual version for a target view or a source view. Figure 1 shows a graphical version of target view for a car application, including object and relationship sets and cardinality constraints. The graphical version does not include the data frames associated with object sets. Figure 2 shows a partial textual version of target view for the car application, including object and relationship sets and cardinality constraints (lines 1-8) and a few lines of the data frames (lines 9-16).

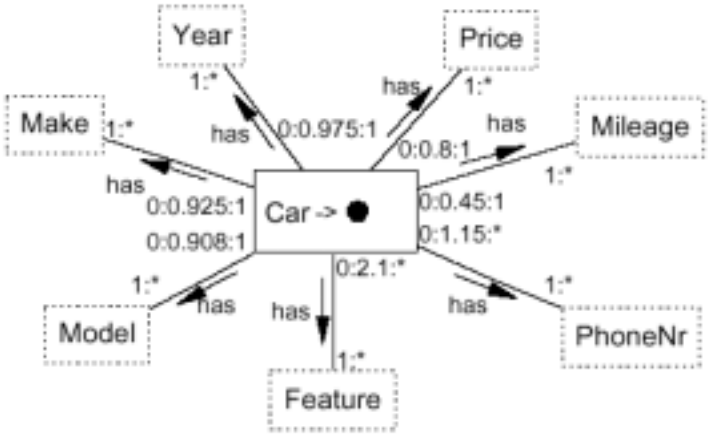

Figure 1: The Graphic Version of Target View For Car Application

```
2. Car [0:0.975:1] has Year [1:*];
3. Car [0:0.925:1] has Make [1:*];
4. Car [0:0.908:1] has Model [1:*];
5. Car [0:0.45:1] has Mileage [1:*];
6. Car [0:2.1:*] has Feature [1:*];
7. Car [0:0.8:1] has Price [1:*];
8. PhoneNr [1:*] is for Car [0:1.15:*];
9. Year matches [4]
10.       constant {extract "\d{2}";
11.             context "\b'[4-9]\d\b";
12.             substitute "^" -> "19"
13.             ...
14. Mileage matches [8]
15.       ...
16.       keyword "\bmiles\b", "\bmi\.", "\bmi\b",
17.             "\bmileage\b";
18. ...
```

Figure 2: The Textual Version of Target View For Car Application (partial)

3.1 Source Discovery

In order to collect applicable information sources for a target view, we first determine if the available information sources (or any portion of the sources) are relevant to the target view. We then analyze the relevant information sources to locate the data that is of interest and reconfigure that data if necessary.

Every target view declares a primary object set of interest such as Car in Figure 1. If an information source contains only one object of interest (e.g., if a Web page only describes a single car), we call it a *single-record* document; if an information source contains two or more objects of interest, we call it a *multiple-record* document. For either a single-record document or a multiple-record document, we call it an *applicable* information source to our target view. If an information source does not contain any useful data of our interest, we call it an *inapplicable* information source to our target view. A document may contain structured data, semi-structured data, or unstructured data. For both single-record and multiple-record documents, each "record" contains the information about an object of interest. The record is composed of the data for a set of related objects and relationships, which are instances of object sets and relationship sets specified in the target view.

Figure 3 shows an example of an applicable information source with respect to a desired target view for the car application shown in Figure 1 and Figure 2. The applicable information source is a multiple-record document containing four records, and each of the records is composed of data compatible with specifications of the object sets including Year, Price, Mileage, PhoneNr, Feature, Model and Make, and the relationship sets including Car has Year, Car has Price, Car has Mileage, Car has PhoneNr, Car has Feature, Car has Model, and Car has Make.

3.1.1 Source Applicability

[Chakrabarti99] points out that crawlers and search engines today do not provide adequate support for automatic source discovery. Though we still can apply crawlers or search engines to selectively seek out documents that may be relevant to the general topic embodied in a target view, we really do not know about the finer level of applicability of each document in the collection to the target view because of the limitations of current crawlers and search engines [BR99]. One way to further filter the documents is to exploit a set of applicability measurements based on the comparison between a document in the collection and the target-view application model. We have applied several applicability measurements including (1) vector space modeling (VSM)[ENX01], (2) logistic regression [Wang01], and (3) multi-variate analysis [Tang01]. Further, we can also apply learning methods [Mitchell97] to automatically select and combine available applicability measures. These learning methods are likely to be particularly useful in an open and evolving information environment such as the Web or for dealing with many different target views.

Figure 3: An Applicable Information Source for Car Application

Usually, applicability measurements depend on data contained in source documents, and the various kinds of documents do make some difference. For a normal unstructured document, the information we are interested in is the raw text contained in the document. With the growing trends of using databases and forms to provide information through the Web, more than 80% of the information [GLSR00] can now only be obtained by a user who fills in a form, which acts as an interface between the user and the serving database. The data behind the forms can only be retrieved with queries. [Yau00] introduces a method to automatically recognize forms and retrieve data behind a Web form. Thus, in addition to the data behind forms, we can also use form fields and database attributes in applicability measurements.

For semistructured data, mainly XML pages, we know that XML describes data with potentially meaningful tags. Thus, in addition to the data contained in an XML document, the content-based tags give us another clue to compute applicability measures.

3.1.2 Source Analysis

Given an applicable information source, if it only contains unstructured data, we would like to apply an ontology-based data extraction technique [ECJL+99] to obtain data from the information source so that the data can be further merged into the target view. However, the applied data extraction technique works well only if the text for each record in the applicable information source can be located and isolated. Thus, we would like to locate the objects of interest and configure them if necessary in the document. Moreover, for a multiple-record or single-record document, although it may be classified as applicable to a target view, part of the document may be inapplicable to the target view. Thus, we need to identify relevant data and discard inapplicable data such as headers and other sections that have nothing to do with the object of interest.

We differentiate between analyzing a multiple-record document and analyzing a single-record document. To locate the objects of interest in a multiple-record Web document, we must determine whether the document is (1) pure (consists only of records of interest), (2) interspersed (consists of applicable and inapplicable records) and (3) linked (has off-page information needed to complete records). If it is necessary to configure the records in a multiple-record Web document, we must determine whether the records are (4) split across natural boundaries, (5) factored with header or trailer data, and/or (6) grouped within natural boundaries. For a single-record document we must only determine whether it is (1) pure, (2) linked and/or (3) split.

Once we determine where the information is and how it is configured we may need to rearrange it. Thus, we locate the records containing the data of interest in the document and discard other inappropriate records. We locate and combine any additional off-page information to make complete records. We put the components of split records together and separate grouped records. And we find the factored information and distribute it appropriately into the records.

3.2 Source/Target Matching

To produce a target-to-source mapping between a target view and a source view, we must respectively match object and relationship sets in the target view with existing or derived object and relationship sets in the source view. If a match potentially relates two existing object sets or relationship sets respectively in a target view and a source view, we can calculate the similarity directly. Otherwise, we can derive object or relationship sets in the source view on the basis of existing object and relationship sets and then can calculate the similarity of derived source object or relationship sets with given target object or relationship sets. The similarity measures determine whether the object sets or relationship sets should match.

3.2.1 Direct Matches

We need to investigate both syntactic and semantic information associated with object or relationship sets to decide whether they match. In our approach, we use different facets of the associated information including the use of (1) names, (2) data

values (associated with either object or relationship sets), (3) context keywords and data descriptions (associated with either object or relationship sets), and (4) constraints (associated with relationship sets).

For names of object and relationship sets, we need a dictionary or thesaurus to obtain potential matches. WordNet [WN, Mil95] is a readily available lexical reference system that organizes English nouns, verbs, adjectives, and adverbs into synonym sets, each representing an underlying lexical concept. We use WordNet as an imported knowledge resource in our approach to construct the metadata and make comparisons between names in source and target views.

For another two facets of metadata, we consider two aspects of data instances associated with object or relationship sets. Since the data instances may have different units and different representations, we provide a set of unit conversion routines and type coercion routines to normalize the data instances so that the data models can be maximally unified. After pre-processing, we can construct a similarity confidence measure based on the characteristics of the data instances. The characteristics gathered depend on the data types: numeric or alphanumeric. For the numeric data type, the characteristics include mean, variance, coefficient of variance, and standard deviation. For the alphanumeric data type, the characteristics include string length, space ratio (number of spaces over the string length), and alphabetic/non-alphabetic ratio (number of alphabetic/non-alphabetic over the string length). A second similarity measure we can construct from the data instances is the use of the expected values in the target view. We can associate with each object or relationship set in the target view a regular expression that matches values expected to appear in a source object or relationship set. Then, using techniques described in [ECJL+99], we can extract values from sources and categorize them with respect to the object or relationship sets in the target view.

Using context keywords and descriptions is another way to measure the similarity. For each object and relationship set in a target view, a data frame may declare its context keywords. We can check for the presence of expected context keywords in data descriptions in an information source.

Figure 4 shows an example of direct matches between object sets of a desired target view for a car application and object sets of a source view extracted from an information source. Both the target view and the source view are shown in their graphical versions. Based on our direct matching technique, we identify the object set matches, including (Car, Car), (Make, Make), (Model, Model), (Year, Year), and (Mileage, Miles). For each of the matches, the first element is the object set in the target view, and the second element is the object set in the source view. We can see that the output does not include the object set pair (Cost, Cost). The reason is that the semantic meaning of “Cost” in the target view is the price of a used car, however, the semantic meaning of “Cost” in the source view is the monthly lease of renting a car. Because our approach considers not only the schematic conflicts such as names of object sets, but also data information associated with the object sets, the generated mapping discards the match (Cost, Cost) because the characteristics of data values associated with the object sets distinguish the semantic meanings.

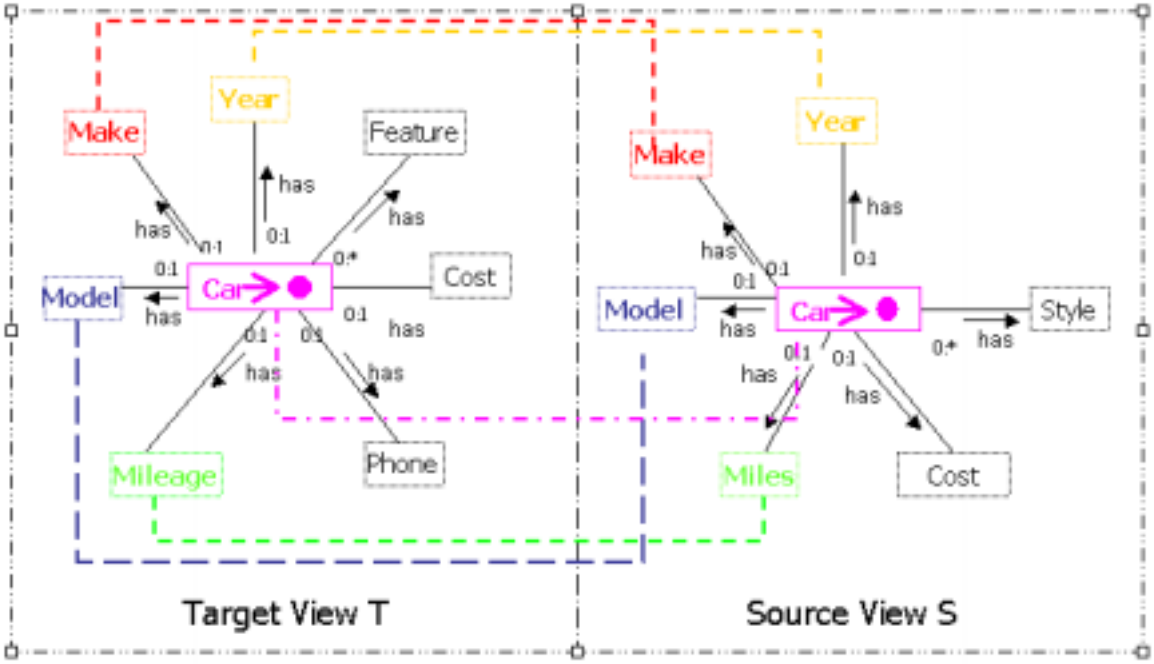

Figure 4: Direct Matches Between Object-sets

Our approach also generates relationship-set matches on the basis of relationship-set constraints. Constraint requirements for relationship sets fall into two basic categories: (1) type requirements needed to satisfy referential-integrity constraints and (2) predicate-calculus constraints. To load a target relationship set from a source relationship set, the type of source objects connecting with the source relationship set must coerce to the type of the target object sets connecting with the target relationship set. We thus must reconcile any discrepancies. If the predicate-calculus constraints on a target relationship set are not coincident with predicate-calculus constraints on a potentially matching source relationship set, the data instances from the source relationship may or may not be directly loadable into a target relationship set. In any case, constraint mismatches make the match suspicious and should be reconciled.

3.2.2 Indirect Matches

Although a source view may not have an object or relationship set that directly corresponds to a declared object or relationship set in a target view, we may be able to derive an object or relationship set that does correspond. In general, we can specify these object- and relationship-set derivations as queries. For example, we can generate a relationship set by means of joining relationship sets along a path in a source view. However, since the number of queries over a view instance is typically unbounded, we are selective in the kinds of queries we generate. The categories of query transformations we consider are: (1) object-set name as value, (2) value aggregation and decomposition, (3) generalizations and specializations of object sets, and (4) path queries including queries over degenerate paths, consisting of only one edge. For each of these transformations we must (1) recognize that we need for the transformation, (2) formulate and translate the transformation query, and (3) derive the constraints for the view

generated by the transformation query. As long as we construct the derived object or relationship sets from specific queries, we are able to derive object instances for the target view from source-view instances.

Figure 5 shows an example of indirect matches between object sets of a target view for a car application and object sets of a source view extracted from an information source. Based on the direct matching technique proposed previously, we can identify the object set matches (Car, Car), (Year, Year) and (Mileage, Miles). However, this output is incomplete since there exist other potential matches between the source view and the target view. For example, if we derive a new virtual object set by aggregating the values associated with two object sets Make and Model in the source view based on the relationships between the object sets Make, Model, and Car, the new virtual object set can match the target object set Make & Model.

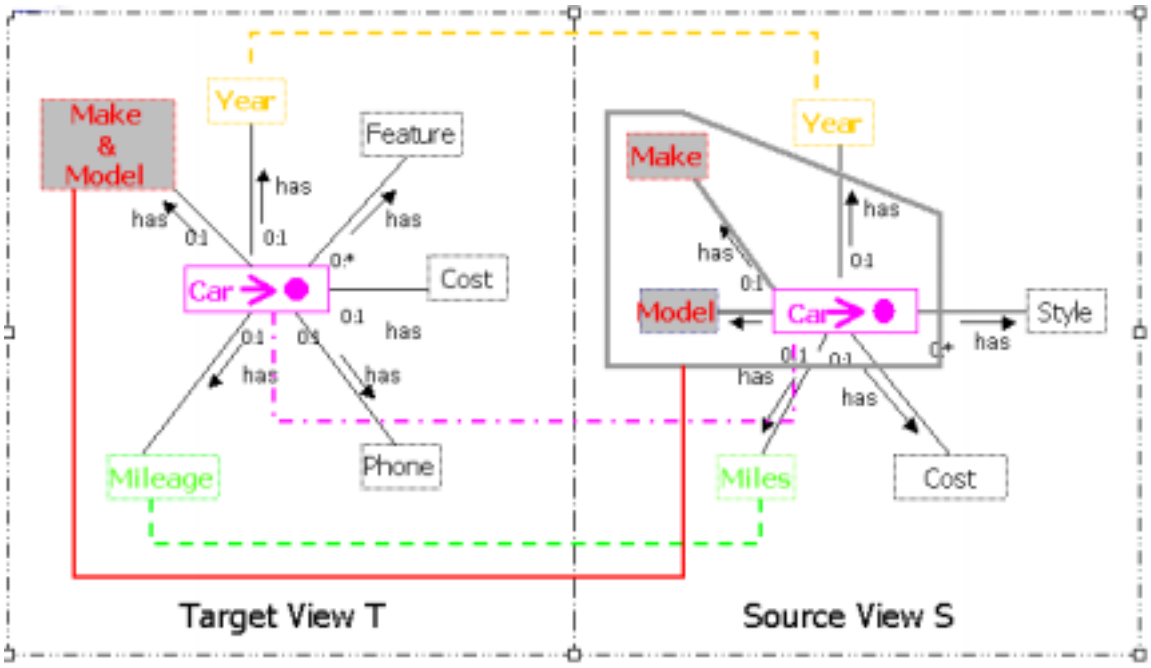

Figure 5: Indirect Matches between Object sets

3.2.3 Similarity Measurement

To exploit the multiple facets of metadata we discussed previously, we build a framework to calculate the similarity between two object sets or two relationship sets. First, we apply each individual, independent facet of metadata to calculate the similarities of pairs of object or relationship sets, one from a target view and the other from a source view. Then, using the confidences of similarity measures obtained from the first step, we combine the similarity measurements for each potential match into a unified measure of similarity.

Although we probably have some idea about what metadata is most useful and in what combination and under what circumstances we should use this metadata, we probably do not know with certainty. Thus, rather than try to encode algorithms over the metadata ourselves, we largely use machine learning to develop the algorithms. This approach also has the advantage of being flexible in the presence of dynamic information sources, which are so common on the Web.

4 Research Plan

We plan to build a demo, run experiments, and publish the results. The demo will be a prototype system to show both our source discovery and source matching work. The experiments will be run against actual Web pages found on the global Internet.

4.1 Demo

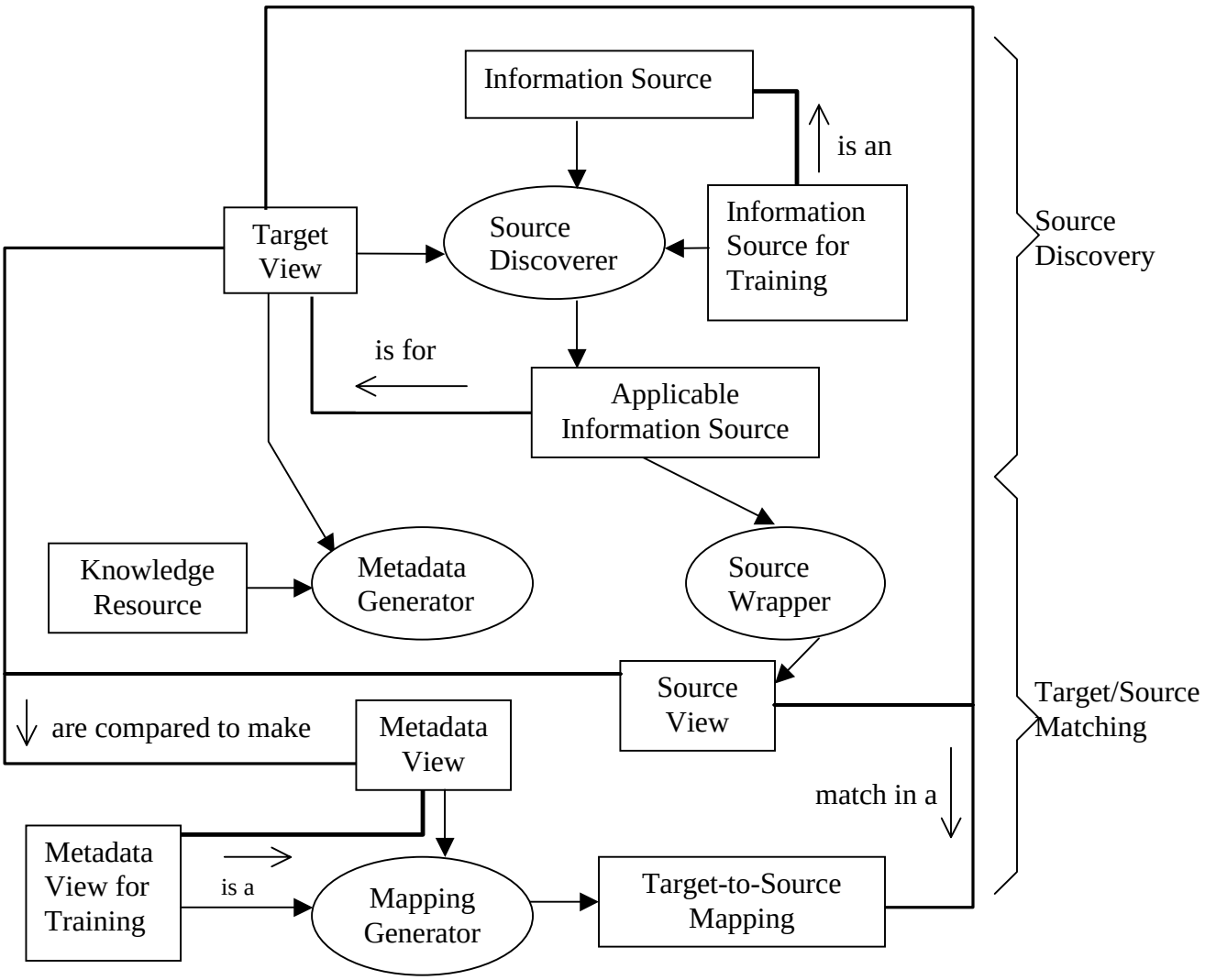

Figure 6: Architecture of Information Integration System

Figure 6 shows the architecture of the demo that implements the techniques discussed in this dissertation. The goal of the architecture is two-fold (1) *Discovery*. The discovery part of the system recognizes and possibly rearranges data from an information source based on the specification of the target view. As part of the discovery process, a set of pre-learned rules decides whether an information source applies to the target view. (2) *Matching*. For an information source with a source view, the matching part of the demo extracts metadata and uses it to generate target-to-source mappings.

In Figure 6, the ovals represent processes that get input resources from incoming arrows and generate outputs along the outgoing arrows (e.g., we can produce applicable information sources through a source-discoverer process for a target view while inputting information sources if training information sources are available.) The boxes represent resources that either act as input or output for processes. The lines that connect boxes and that have labeled reading direction arrows represent relationships among resources (e.g., the line between the boxes denoted as "Applicable Information Source" and "Target View" represents the relationship "Applicable Information Source is for Target View.") We describe each component of the system in the following.

- Target View. A target view is an ontology [ECJL+99], a conceptual-model instance that describes a real-world application in a narrow, data-rich domain of interest. We model target views in OSM [EKW92].
- Information Source. An information source is any one of a set of Web pages collected into the system. The information sources can be database tables, semistructured documents, or unstructured documents.
- Information Source for Training. The set of information sources for training is a subset of the information sources. We apply machine-learning algorithms to the training set to generate rules to decide whether an information source applies based on a target view. The training set is composed of positive examples and negative examples designated by a domain expert.
- Source Discoverer. The source discoverer recognizes and possibly rearranges objects and relationships according to the declarations in a target view. The source discoverer uses data from training sources to obtain a set of decision rules based on available applicability measurements. It then applies the decision rules to determine if an information source applies to a target view.
- Applicable Information Source. Each applicable information source applies to one of target views stored in the system.
- Source Wrapper. The source wrapper transforms an information source into a populated OSM model instance. The wrapper generator partially relies on existing middleware tools.
- Source View. The source views are OSM model instances populated with source data.
- Metadata Generator. The metadata generator obtains metadata from a target view and source views and generates metadata views with the help of knowledge resources.
- Knowledge Resource. The knowledge resources are external resources that contain useful knowledge. WordNet, for example, is one of the imported knowledge resources.
- Metadata View. The metadata view consists of the various abstractions of the metadata used to describe the similarity between a target view and a source view. The metadata view provides a framework for multifaceted exploitation of metadata in which we gather information about potential matches from various facets of metadata and make it possible to combine this information to generate and place confidence values on potential concept matches.
- Metadata View for Training. The document abstraction in the metadata view for training is a subset of the abstraction in the metadata view. We use the training

set to apply machine-learning algorithms to generate rules to decide whether a target object set can match an actual or virtual source object set or a target relationship set can match an actual or a virtual source relationship set.

- Mapping Generator. The mapping generator generates target-to-source mappings between a source view and a target view. The mapping generation is based on a set of pre-trained decision rules.
- Target-to-Source Mapping. The target-to-source mappings consist of matched object sets and matched relationship sets between a target view and a source view. Source-view object and relationship sets may be virtual, being derived as queries over actual source object and relationship sets.

4.2 Experiments

We will run several experiments to determine the effectiveness of our approach on real data. When we choose the applications for our experiment, we would like to select applications that represent a rich set of domains, so that we can evaluate whether the approach is reliable and has high performance over a wide spectrum of applications. We divide the domain data into three partitions with respect to their functionalities: (1) business data, (2) historical data, and (3) scientific data.

Based on the data partitions, at least three applications will be considered: (1) buying and selling (automobiles as an example)—business data, (2) genealogy—historical data, and (3) biological structure—scientific data, but we intend to consider other applications so long as sufficient data exists. For each task, based on the proposed architecture, we distinguish between two different phases: (1) a training phase for semi-automatically generating source discovery and target/source matching rules, and (2) a testing phase for judging the applicability of source views and for producing target-to-source mappings. We will record both the training times and the testing times for each application and analyze applied algorithm complexities based on the application model and the performance computations. Besides this, several human experts will evaluate the test results. Assuming the human experts are correct, the three measures including *accuracy*, *precision*, and *recall* will be analyzed and discussed. Assuming that

(1) *T* is the set of all applicable information sources obtained for a desired target view or generated target-source matches between a target view and a source view by our approach and

(2) *T'* is the set of all inapplicable information sources obtained for a desired target view or incorrectly generated target-source matches between a target view and a source view by our approach and

(3) *L* is the set of all application information source for a desired target view or target-source matches between a target view and a source view for the test phase and

(4) *L'* is the set of all inapplicable information source for a desired target view or inappropriate target-source matches between a target view and a source view for the test phase and

(5) *TL* is the set of correctly classified applicable information sources and correctly generated target-source matches and

(6) *T'L'* is the set of correctly classified inapplicable information sources and correctly generated inappropriate target-source matches

then accuracy, precision and recall are obtained as follows

$$accuracy = (TL + T'L')/(L + L')$$

$$precision = TL/T$$

$$recall = TL/L$$

4.3 Applicable Machine-Learning Techniques

In the framework of our approach, we largely apply learning-based algorithms to explore available quantitative measures and learn the discovery and matching algorithms. We compute the quantitative measures through data analysis and schema analysis techniques. Our framework includes five learning techniques and an ad-hoc heuristics technique as potential resolutions to the six issues as shown in Table 1. As we proceed, we will further analyze the potential resolutions to determine a final resolution to an issue.

	DT	kNN	NB	RL	HMM	AH
Source Discovery (data)	*	*	*			*
Source Discovery (data + schema)	*	*	*			*
Direct Object-Set Matching	*	*	*			*
Direct Relationship-Set Matching	*	*	*			*
Indirect Object-Set Matching	*	*	*	*	*	*
Indirect Relationship-Set Matching	*	*	*	*	*	*

Note: DT (Decision Tree), kNN (k-Nearest Neighbor), NB (Na?ve Bayesian), RL (Reinforcement Learning), HMM (Hidden Markov Modeling), and AH (Ad Hoc Heuristics).
Table 1: Research Issues and Possible Resolutions

In Table 1, every cell corresponds an <i: issue, r: resolution> pair. If a "*" appears in the cell, it means that we can apply the resolution r to the issue i.

We first discuss the basic ideas of the learning techniques and then explain how they apply to the discovery and matching issues.

4.3.1 Decision Tree (DT), k-Nearest Neighbor (kNN) and Na?ve Bayesian (NB)

DT, kNN and NB learning algorithms apply to learning tasks where each instance *x* is described by a conjunction of attribute values $A < a_1, a_2, ..., a_n >$ and where the target function *f(x)* can take on any value from some finite set $V < v_1, v_2, ..., v_m >$.

Decision tree (DT) learning applies the C4.5 algorithm [Mitchell97], which is an active learning algorithm. The output of DT learning is a decision tree. The central choice in the C4.5 algorithm is selecting which attribute to test at each node in the tree. We would like to select the attribute that is most useful for classifying examples. In order to find a good quantitative measure of the worth of an attribute, the algorithm

defines a statistical property, called *information gain* denoted by $Gain(S, a_i)$, which measures how well a given attribute $a_i (1 ? i ? n)$ separates the training examples *S* according to their target classification. Formula 1 shows that the C4.5 algorithm uses this information-gain measure to select an attribute a_{MAP} that has the best information gain among the candidate attributes at each step while growing the tree, where argmax means "average maximum."

$$a_{MAP} = \arg\max_{a_i ? A} Gain(S, a_i) \quad (1)$$

k-Nearest Neighbor (kNN) [Mitchell97] learning is a lazy learning algorithm. This algorithm assumes that all instances, $< a_1, a_2, ..., a_n >$, correspond to points in an *n*-dimensional space. The nearest neighbors of an instance are defined in terms of the standard Euclidean distance.

The Na?ve Bayesian (NB) learning algorithm is based on the simplifying assumption that the attribute values are conditionally independent given the target value. Formula 2 shows that NB classifies a new instance as the most probable target value, v_{MAP}, given the attribute values $< a_1, a_2, ..., a_n >$ that describe the instance.

$$v_{MAP} = \arg\max_{v_j ? V} P(v_j) ? \prod_i P(a_i \mid v_j) \quad (2)$$

where v_{MAP} denotes the target value output by the NB learning algorithm, $P(v_j)$ is the probability of the target value v_j, and $P(a_i \mid v_j)$ is the conditional probability of a_i for target value v_j.

4.3.2 Reinforcement Learning (RL)

Reinforcement Learning (RL) [Mitchell97] applies the *Q* learning algorithm [Mitchell97] to learn to control a sequential process for a special issue. RL addresses how an agent learns control policies by means of trial-and-error interactions with a dynamic environment. One important feature that makes RL unique is that it provides a way for an agent to learn by rewards and punishment without needing to specify how the task is to be achieved. In RL, a task is defined by a set of states *S*, a set of actions *A*, and a state-action transition function $\delta : S?A->S$. At each time step, the learner selects an action, and then as a result is given a reward and its new state. The goal of reinforcement learning is to learn a policy, a mapping from states to actions: $\pi: S->A$ that maximizes the sum of its reward over time.

In the *Q* learning algorithm, the agent can learn the evaluation function *Q(s,a)*, which is shown in Formula 3, so that its value is the maximum discounted cumulative reward that can be achieved starting from state *s* and applying a particular action *a* as the first action.

$$Q(s, a) ? r(s, a) + \gamma V^*(\delta(s, a)) \quad (3)$$

where r is the immediate reward by producing the succeeding state $\delta(s,a)$, $0 ? \gamma < 1$ is a constant that determines the relative value of delayed versus immediate rewards, and V^*

is defined to be the sum of discounted future rewards over the infinite future. The goal of reinforcement learning is to find the policy

$$\pi(s) = \arg\max_a Q(s,a)$$

4.3.3 Hidden Markov Modeling (HMM)

Hidden Markov Modeling (HMM) is composed of a set of states *Q*, with specified initial and final states q_0 and q_F, a set of transitions between states ($q_0 \overset{*}{?} q_F$), and a discrete vocabulary of output symbols $\Sigma = \sigma 1 \sigma 2 \ldots \sigma M$ [RJ86]. The model generates a string $x = x1x2\ldots xi$ by beginning in the initial state, transitioning to a new state, emitting an output symbol, transitioning to another state, emitting another symbol, and so on, until a transition is made into the final state. We define *P(q->q')* as the transition probability that one state *q* follows another state *q'*, and *P(q? σ)* the probability that a state *q* emits a particular output symbol σ. Once these are given, the probability of a particular path through the model generating the string *x* can be computed as the product of all transition and emission probabilities along the path. The probability of the string *x* is the sum of the probabilities of all paths generating *x*.

4.3.4 Ad Hoc Heuristics

For this technique, we apply heuristic rules to work on the issues listed in Table 1.

4.3.5 Application of Techniques to Research Issues

Every cell in Table 1 corresponds to a pair composed of an issue and a resolution. Since we have already described the basic ideas of the resolutions, we now focus on how to apply the techniques to resolve the issues.

4.3.5.1 DT, KNN, and NB Cells

For each of the issues listed in the first column of the table, we can apply DT, kNN, and NB to resolve the issue. During training phase, we are given a set of training documents and the target function *f(x)* for each document *x*. For each training document *x*, we can compute a set of similarity measures representing semantic similarities and structural similarities between a target view and information sources. In order to compute the similarity measures, we can consider several aspects related to data and schema characteristics such as means of numeric values, lengths of alphanumeric values, and synonyms of names in schemas. One measure corresponds to each attribute $a_i (1 ? i ? n)$ in $< a_1, a_2, ..., a_n >$. Thus, we can collect a set of tuples as training data for the three learning techniques. Each tuple, according to a document *x*, contains a set of measures and a target value $< a_1, a_2, ..., a_n, f(x) >$. These tuples constitute the training data. During the testing phase, given a new instance *x*, we can classify it based on a decision tree output by DT, or based on the properties of the instance's nearest neighbors stored in training data by kNN, or based on the most probable target value v_{MAP} calculated by NB.

When we calculate the attribute values for the instances input to the learning techniques, the data used for training and testing can contain errors—either the attribute values have errors since the measurement calculation sometimes depends on ad-hoc

heuristic techniques or the target values have errors since the domain expert may inadvertently introduce error target values. A key feature of DT learning is that it is one of the best learning algorithms for a small set of training data and it is robust against noisy training data. The kNN learning module is also robust against noisy training data. A practical issue in applying kNN learning is that the distance between instances is calculated based on all attribute values of the instances. This lies in contrast to methods such as DT learning, which selects only a subset of instance attributes when forming the decision tree. Thus, by applying the kNN learning module, we can consider each kind of semantic or structural conflicts when testing an instance and we can weight the attribute values to differentiate the importance of the measures. NB learning is useful in many practical applications, even when the simple assumption of NB learning is not met.

4.3.5.2 RL and HMM Cells

These cells in Table 1 show that we apply HMM and RL to construct virtual object sets/relationship sets for indirect object-set/relationship-set matching. The virtual object sets and the virtual relationship sets are views produced by queries over existing object sets or relationship sets in source views. When a new information source enters the system, instead of exhaustively exploring the queries or manually constructing the queries, HMM and RL can recognize and produce the virtual object sets or virtual relationship sets automatically. Given the virtual object sets or virtual relationship sets, we can further apply ad hoc heuristics or DT, kNN, and NB to resolve mappings over derived source views.

Setting up RL or HMM learning requires two inputs: (1) a target view that has both schema and data instances, perhaps given as regular expressions, and (2) a source view that has both schema and data instances, perhaps generated by wrapper tools. The state set *S* of an RL or HMM learner corresponds to the set of object sets in the source view with respect to a goal state in the target view, and the state transitions correspond to the relationship sets between the object sets in the view. The output symbol of the state in HMM learning is defined as the set of data instances associated with the object set in the source view. The reward of RL learning is given based on similarity calculation over both data and schema between the current state and the goal state.

The key features of virtual object-set or virtual relationship-set construction that make RL the proper learning method for defining an optimal solution are: (1) performance is measured in terms of reward over time, and (2) the environment presents situations with a delayed reward. HMM learning also provides us an applicable method for virtual object-set or virtual relationship-set construction. HMM was originally used in speech recognition [HAJ90] and it has become the most successful speech model. The main reason for this success is its ability to characterize the speech signal in a mathematically tractable way. A well-defined formalism exists, which helps with the theoretical understanding of what can be expected when applying it to sequence analysis. Also, Bayesian statistics is used in several aspects of the method. Given a sequence of multiple object sets and relationship sets, we can use statistical methods to build a specific HMM. The probabilities that are required are estimated from the similarity measures. Thus, HMM can be used to test other sequences whether they match or not.

4.3.5.3 Ad Hoc Cells

For each of the listed issues, we can apply ad-hoc heuristics to resolve it. We prefer learning methods, but may need to resort to ad-hoc techniques.

4.4 Delimitations

In this dissertation, the following issues will not be addressed.

- Merging related data from applicable information sources in target views.
- Creating new wrapper generation tools.

5 Research Papers

- ? Locating and Reconfiguring Records in Unstructured Multiple-Record Web Documents
- ? Source Discovery for Multiple-Record Information Sources
- ? Source Discovery Techniques for Multiple Types of Knowledge Sources
- ? Exploitation of Metadata for Attribute Matching in Information Integration
- ? Generating Direct Target-to-Source Mappings from an Information Source to a Target View
- ? Generating Indirect Target-to-Source Mappings from an Information Source to a Target View
- ? An Environment for Target-Based Source Discovery and Source Matching

6 Contribution to Computer Science

Our vision is to provide a truly generic and powerful architecture to enable quality source discovery and source matching based on a predefined target view. The target view is independent of any available information source, which makes the approach scalable in open and evolving environments. The architecture provides a framework, which largely applies machine-learning techniques to learn the source-discovering and source-matching algorithms. These machine-learning algorithms exploit a set of available metadata associated with the target view and the information sources, instead of applying hard-coded heuristics constructed manually by end users. The developed system should classify, and rearrange if necessary, the applicable information in sources for the target view and should generate the target-to-source mappings. It is intended that a prototype implementation will evaluate and validate the ideas described in this proposal. The prototype will also provide a graphical user interface for the user and should enable the discovery and matching procedures to be semi-automatic, requiring as little user's intervention as possible.

7 Dissertation Schedule

? Locating and Reconfiguring Records in Unstructured Multiple-Record Web Documents (Dec., 2000)

? Source discovery for multiple knowledge source integration (Dec. 2001)

- o Source analysis
  - Unstructured (Dec., 2000)
  - Structured (Oct., 2001)
  - Semi-structured (Oct., 2001)
- o Analysis method combination (Dec., 2001)

? Extracting Target-to-Source Mappings from Information Sources into a Target View (Aug., 2002)

- o Direct matching (Apr., 2002)
  - Concept matching
  - Relationship matching
- o Indirect matching (Dec., 2002)
  - Evaluation of necessity
  - Concept and relationship restructuring
  - Matching techniques

? An Example-based Environment for Target Based Source Discovery and Source Matching (Apr., 2003)

- o Framework
- o Demo implementation

8 References

[Abelberg98] B. Abelberg. NoDoSE - A Tool for Semi-Automatically Extracting Structured and Semistructured Data from Text Documents. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Pages 283—294, Seattle, Washington, 1998.

[ACHK93] Y. Arens, C.Y. Chee, C. Hsu, C.A. Knoblock, Retrieving and Integrating Data from Multiple Information Sources, *International Journal of Intelligent and Cooperative Information Systems*, Vol. 2, No. 2, Pages 127—158, 1993.

[BB63] H. Borko and M. Bernick. Automatic Document Classification. *Jounal of ACM*, Vol. 10, No. 2, Pages 151—162, 1963.

[BCV99] S. Bergamaschi, S. Castano, and M. Vincini. Semantic integraion of semistructureed and structured data sources. *SIGMOD Record*, Vol. 28, No. 1, Pages 54—59, March 1999.

[BGLM+99] C. Baru, A. Gupta, B. Ludascher, R. Marciano, Y. Papakonstantinou, P. Velikhov, and V. Chu. XML-Based Information Mediation with MIX. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Pages 597—599, Philadelphia, 1999,

[BR99] R. Baeza-Yates, and B. Ribeiro-Neto. *Modern Information Retrieval*, Addison Wesley, England, 1999.

[CA99] S. Castano and V.D. Antonellis. A schema analysis and reconciliation tool environment for heterogeneous databases. In *Proceeds of the International Database Engineering and Applications Symposium (IDEAS)*, Pages 53—62, 1999.

[CAFP98] S. Castano, V.De Antonellis, M.G. Fugini, and B. Pernici. Conceptual schema analysis: techniques and applications. *ACM Transactions on Database systems*, Vol. 23, No. 3, Pages 286—333, September 1998.

[Chakrabarti99] S. Chakrabarti. Recent results in automatic Web resource discovery, *ACM Computing Surveys*, Vol. 31, No. 4es, December 1999.

[CHR97] C. Clifton, E. Housman, and A. Rosenthal. Experience with a Combined Approach to Attribute-Matching across Heterogeneous Databases. In *Proceedings of the IFIP Working Conference on Data Semantics (DS-7)*, 1997.

[CHSA+95] M. J. Carey, L.M. Haas, P. M. Schwarz, M. Arya, W.F. Cody, R. Fagin, M. Flickner, A.W. Luniewski, Wayne Niblack, D. Petkovic, J. Thomas, J.H. Williams and E.L. Wimmers. Towards Heterogeneous Multimedia Information Systems: The Garlic

Approach. *Proceedings of the Fifth International Workshop on Research Issues in Data Engineering (RIDE): Distributed Object Management*, 1995.

[DDL00] A. Doan, P. Domingos, A. Levy. Learning Source Descriptions for Data Integration. *Third International Workshop on the Web and Databases*, pages 81—92, Dallas, Texas, May, 2000.

[DDH01] A. Doan, P. Domingos, A. Halevy. Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, May 21-24, Santa Barbara, California, 2001.

[ECJL+99] D.W. Embley, E.M. Campbell, Y.S. Jiang, S.W. Liddle, D.W. Lonsdale, Y.-K. Ng, and R.D. Smith. Conceptual-Model-Based Data Extraction from Multiple-Record Web Documents, *Data and Knowledge Engineering*, November, 1999.

[EJN99] D.W. Embley, Y.S. Jiang, and W.-K. Ng. Record-Boundary Discovery in Web Documents. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Philadelphia, 1999.

[EKW92] D.W. Embley, B.D. Kurtz, and S.N. Woodfield. *Object-oriented Systems Analysis: A Model-Driven Approach.* Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[ENX01] D.W. Embley, Y.-K. Ng, and L. Xu. Recognizing Ontology-Applicable Multiple-Record Web Documents, appear on ER2001.

[FB92] W.B. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithsms*. Prentice-Hall, 1992.

[FPNB99] J. Fowler, B. Perry, M. Nodine, and B. Bargmeyer. Agent-based semantic interoperability in InfoSleuth. *SIGMOD Record*, Vol. 28, No. 1, Pages 60—67, March 1999.

[GLSR00] P.B. Golgher, A.H.F. Laender, A.S.da Silva, B. Ribeiro-Neto. An Example-Based Environment for Wrapper Generation. *Second International Workshop on the World Wide Web and Conceptual Modeling (WCM2000)*, Pages 152—164, Salt Lake City, Utah, October, 2000.

[Goodman91] M. Goodman. Prism: A Case-based Telex Classifier. In *Proceedings of the 2nd Annual Conference on Innovative Applications of Artificial Intelligence. AAAI Press*, Pages 25—37, Menlo Park, California, 1991.

[GLY99] H.Garcia-Molina, W. Labio, and R. Yerneni. Capability sensitive query processing on internet sources. *In Proceedings of the 15th International Conference on Data Engineering*, Sydney, Australia , March, 1999.

[GKD97] M. Genesereth, A. Keller, and O. Duschka. Infomaster: An Information Integration System, In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Arizona, 1997.

[GPQR+97] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J.D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages. *Journal of Intelligent Information Systems*, Vol. 8, No. 2, Pages117—132, 1997.

[HAJ90] X.D. Huang, Y. Ariki, and M.A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.

[Hammer99] J. Hammer. The Information Integration Wizard (Iwiz) Project Report Work in Progress, Univ. of Florida, Technical Report, TR99-019, Gainesville, FL 32611-6120.

[Hoyle73] W. Hoyle. Automatic Indexing and Generation of Classification Systems by Algorithms. *Information Storage Retrieval*, Vol. 9, No. 4, Pages 233—242, 1973.

[Hul97] R. Hull. Managing Semantic Heterogeneity in Databases: A Theoretical Perspective. In *Proceedings of the Sixteenth ACM SIG-SIGMODSIGART Symposium on Principles of Database Systems*, May 12-14, 1997, Tucson, Arizona, Pages 51-61, New York, NY 10036, USA, 1997. ACM Press.

[HW91] P. J. Hayes and S. P. Weinstein. Construe-TIS: A System for Content-based Indexing of A Database of News Stores. In *Proceedings of the 2nd Annual Conference on INNOVATIVE Applications of Artificial Intelligence. AAAI Press*, Pages 49—64, Menlo Park, California, 1991.

[HZ96] R. Hull, and G. Zhou. A Framework for Supporting Data Integration Using the Materialized and Virtual Approaches. In *Proceedings ACM SIGMOD Conference on the Management of Data***,** Pages 481—496, Montreal, Canada, 1996.

[IFF99] Z. Ives, D. Florescu and M. Friedman. An adaptive Query Execution System for Data Integration. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Pages 299—310, *Philadelphia*, *1999*.

[LAW] T. Lahiri, S. Abiteboul, and J. Widom. Ozone: Integrating Structured and Semistructured Data. http://wwwdb. stanford.edu/pub/papers/ozone.ps

[LC00] W. Li, and C. Clifton. SEMINT: A Tool for Identifying Attribute Correspondence in Heterogeneous Databases Using Neural Networks. *Data and Knowledge Engineering*, Vol. 33, No. 49, Pages49—84, 2000.

[LMSS95] J.J. Lu, G. Moerkotte, J. Schue, and V.S. Subrahmanian. Efficient Maintenance of Materialized Mediated Views. In *Proceedings of ACM SIGMOD Conference on the Management of Data*, pages 340—351, 1995.

[KS98] V. Kashyap and A. Sheth. Semantic heterogeneity in global information systems: The role of metadata, context and ontologies. In M. Papazoglou and G. Schlageter, editors, *Cooperative Information Systems: Current Trends and Dictions*, pages 138—178, 1998.

[Maron61] M. Maron. Automatic Indexing: An Experimental Inquiry. *Journal of ACM*, Vol. 8, Pages 404—417, 1961.

[MBFG+93] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross, K. Miller, and R. Tengi. Five papers on WordNet, Princeton University, [Online: ftp://fp.cogsci.princeton.edu/pub/wordnet/5paper.pdf], Auguest, 1993.

[MBR01] J. Madhavan, P.A. Bernstein, and E. Rahm. Generic Schema Matching with Cupid, In *Proceedings of VLDB*, September 11-14, Roma, Italy, 2001.

[MHH00] R. Miller, L. Haas, and M. Hernandez. Schema Mapping as Query Discovery. In *Proceedings of VLDB*, September 10-14, Cairo, Egypt, 2000.

[Mitchell97] T.M. Mitchell. *Machine Learning*, McGraw-Hill Companies, Inc. 1997.

[MKSI96] E. Mena, V. Kashyap, A. Sheth and A. Illarramendi. "OBSERVER: An approach for Query Processing in Global Information Systems based on Interoperation across Pre-existing Ontologies," *In Proceedings of the 1st IFCIS International Conference on Cooperative Information Systems (CoopIS '96), Brussels, Belgium*, June 1996.

[MMN99] I. Muslea and S. Minton and G. Knoblock. A Hierarchical Approach to Wrapper Induction. In *Proceedings of the 3rd Conference on Autonomous Agents* 1999 (1999). http://www.isi.edu/muslea/PS/hwi_aa99.ps

[MZ98] T. Milo and S. Zohar. Using Schema Matching to Simplify Heterogeneous Data Translation. In *Proceedings of VLDB*, August 24-27, New York City, New York, 1998.

[PE95] M. Perkowitz and O. Etzioni. Category Translation: Learning to Understand Information on the Internet. In *Proceedings of International Joint Conference on AI (IJCAI)*, 1995.

[PSBG+99] N.W. Paton, R. Stevens, P. Baker, C.A. Goble, S. Bechhofer, and A. Brass. Query Processing in the TAMBIS Bioinformatics Source Integration System. *In Proceedings of 11th International Conference on Scientific and Statistical Database Management*, pages 138—147, Cleveland, Ohio, 1999.

[PSU98] L. Palopoli, D. Sacca, and D. Ursino. Semi-automatic Semantic Discovery of Properties from Database Schemes. In *Proceedings of the International Database Engineering and Applications Symposium (IDEAS)*, Pages 244—253, 1998.

[RJ86] L.R. Rabiner and B.H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, pages 4—16, January 1986.

[RJ91] L.F. Rau and P.S. Jacobs. Creating Segmented Databases from Free Text for Text Retrieval. In *Proceedings of SIGIR ACM*, Pages 337—346, New York, 1991.

[SA99] A. Sahuguet and F. Azavant. Web Ecoogy: Recycling HTML pages as XML DOCUMENTS USING w4f. In *Proceedings of Second International Workship on the Web and Databases* (1999). http://db.cis.upenn.edu/DL/webdb99.ps.gz.

[Singh98] N. Singh, Unifying Heterogeneous Information Models. *Communications of the ACM*, Vol. 41, No. 5, May 1998.

[SM83] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[SSR94] E. Sciore, M. Siegel and A. Rosenthal. Using Semantic Values to Facilitate Interoperability Among Heterogeneous Information Systems. *ACM Transactions on Database Systems*, Vol. 19, No. 2, Pages 254—290, June 1994.

[Tang01] J. Tang. A Probabilistic Model For Binary Categorization of Multiple-Record Web Documents. A Master degree of Science thesis, Brigham Young University, Provo, Utah, 2001.

[TRV96] A. Tomasic, L. Raschid, and P. Valduriez. Scaling Heterogeneous Databases and the Design of Disco. *International Conference on Distributed Computing Systems*, pages 449—457, 1996.

[Ullman97] J.D. Ullman. Information Integration Using Logical Views. *In Proceedings of the 6th International Conference on Database Theory (ICDT-97), Lecture Notes in Computer Science*, Pages 19—40, Springer-Verlag, 1997

[Wang01] Q. Wang. Ontology-Based Binary Categorization of Multiple-Record Web Documents Using a Probabilistic Retrieval Model, A Master degree of Science thesis, Brigham Young University, Provo, Utah, 2001.

[WN] WordNet home page: http://www.cogsci.princeton.edu/~wn/w3wn.html

[Yau00] S.T. Yau. Automating the Extraction of Data Behind Web Forms, A Master thesis proposal, Brigham Young University, Provo, June 2000.

[ZGHW95] Y. Zhuge, H. Garcia-Molina, J. Hammer, and J. Widom. View Maintenance in a Warehousing Environment. In *Proceedings of ACM SIGMOD Conference on the Management of Data*, Pages 316—327, 1995.

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a dissertation proposal submitted by

Li Xu

This dissertation proposal has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

_____	_____
Date	David W. Embley, Chair
_____	_____
Date	Stephen W. Liddle
_____	_____
Date	Todd Peterson
_____	_____
Date	Dan R. Olsen
_____	_____
Date	Scott Woodfield
_____	_____
Date	David W. Embley, Graduate Coordinator